

A UML and Petri Nets Integrated Modeling Method for Business Processes in Virtual Enterprises

Gou Hongmei, Huang Biqing and Ren Shouju

The State CIMS Engineering and Research Center,
Tsinghua University, Beijing 100084 P.R.China
hmgou@mail.au.tsinghua.edu.cn, hbq@cims.tsinghua.edu.cn

From: AAAI Technical Report SS-00-03. Compilation copyright © 2000, AAAI (www.aaai.org). All rights reserved.

Abstract

Virtual Enterprise is an important organization pattern for future enterprises, one of whose major functions is the distributed and parallel business process execution. This paper aims at the study on business process modeling in virtual enterprises. Based on the object-oriented description of business processes in virtual enterprises, we propose a UML and Petri nets integrated modeling method for business processes in virtual enterprises. The method provides an integrative framework supporting requirement description, model specification and design, model analysis and simulation, and model implementation.

Introduction

The trend towards global markets and the increasing customer orientation gives rise to a new manufacturing paradigm - Virtual Enterprise (VE). A virtual enterprise is a temporary coalition of enterprises that come together to share costs and skills to address business opportunities that they could not undertake individually (NIIP 1998). Nowadays, Virtual Enterprise has been considered as one of the most promising paradigms for future enterprises.

One of VE's major functions is the distributed and parallel business process execution by function entities of each VE member. For supporting this, effective business process modeling with strict model analysis is required urgently. However, such studies for VEs just begin with scarce fruits. Our work presented here aims at the study on business process modeling in VEs. We propose a Unified Modeling Language and Petri Nets (UML-PNs) integrated modeling method based on the Object-Oriented (OO) description of business processes in VEs. The method will provide an integrative framework supporting requirement description, model specification and design, model analysis and simulation, and model implementation.

The rest of this paper is organized as follows. In section 2, we introduce the Unified Modeling Language (UML) and Petri Nets (PNs) briefly and expound the motivation to integrate them. In section 3, we provide an OO description of business processes in VEs, based on which we discuss the UML-PNs integrated method for business processes in VEs in detail. The final section concludes the paper.

Background and Motivation

The UML Emerges from the unification and standardization of OO modeling methods. As a visual modeling language, contents of the UML are embodied as the following diagrams (Rational 1997). *Use case diagram* shows the relationship among actors (roles played by outside objects) and use cases (coherent units of functionality) within a system. *Class diagram* describes the static structural model consisting of model elements, e.g., classes, interfaces and their relationships. *Object diagram* is instance of a class diagram. *Package* is a grouping of model elements. *Statechart diagram* represents a state machine, which is a graph of states and transitions. *Activity diagram* is a variation of a state machine for a procedure, in which states are activities representing operation performances and transitions are triggered by the completion of operations. *Interaction diagram* shows interaction pattern among objects and comes in *sequence diagram* (shows interaction arranged in time sequence with explicit sequence of messages) and *collaboration diagram* (shows interaction organized around objects). *Implementation diagram* shows aspects of implementation and comes in *component diagram* and *deployment diagram*.

Through UML diagrams, we can describe user requirements, static properties and dynamic behaviors of a system in a convenient and straightforward way. It is also easy to transform the UML model to program source codes. However, since the UML is an informal modeling language, it is difficult to analyze the UML model strictly.

Petri Nets (PNs) provide a powerful formal modeling method based on solid mathematical fundament while having graphical representation of system models as net diagrams. The formal definitions of PNs can be found in many references, such as (Murata 1989) and are skipped here due to page limit. A graphical PN model consists of circles, rectangles, directed arcs and dots, which represent places, transitions, arcs and tokens respectively. Besides, PNs provide various analysis techniques such as reachability tree, incidence matrix and invariant analysis method, through which properties of the PN model such as liveness, reachability and deadlock can be analyzed.

Through PNs, we can analyze the structure and dynamic properties of systems through strict mathematics analysis and visualized computer simulation as well as model distributed and parallel processes. However, the graphical representation of PNs is still far from the way by which

people understand and describe real-world systems, bringing difficulties in establishing a PN model.

We can see the UML and PNs have reciprocal features as follows: 1).The UML is user-friendly while PNs possess formal strictness. 2).The UML can describe systems effectively while PNs can analyze systems strictly. 3).The UML model can be implemented easily while the PN model is suitable for simulation. Many researchers has done some work about the integration of OO and PNs, such as (Sibertin-Blanc 1994) (Lakos 1995). Here, we combine advantages of the UML and PNs and propose a UML-PNs integrated modeling method for business processes in VEs.

The UML-PNs Integrated Modeling Method

In this section, we will provide business processes in VEs an object-oriented description, based on which we will elaborate the UML-PNs integrated modeling method following the lifecycle of requirement description, model specification and design, model analysis and simulation, and model implementation. In the discussion, a simplified demonstrative virtual enterprise that embarks on computer manufacturing will be used as an example.

Business Processes in VEs - an OO Description

Generally, a business process can be decomposed into many business activities, which will be assigned to particular task-performing entities. In a virtual enterprise, business activities are distributed in multiple VE members and performed in parallel. There exist logic and temporal relations among these business activities.

Then, we describe business processes in VEs based on OO methodology. Each entity is modeled as an object and we call it *business object* from now on. A business object has its own attributes and operations. Attributes represent its properties (e.g. ID, belonging VE member, information and resources required) while operations represent business activities it can perform. Each operation has its enabled pre-conditions and will produce some post-conditions, which are transferred by message transmission. In conclusion, a virtual enterprise can be modeled as a set of distributed business objects communicating with each other and the interrelated business activities performed by them in parallel constitute the business process of the VE.

Contents of the UML-PNs Modeling Method

The global scheme of the UML-PNs integrated modeling method (Fig.1) can be described as follows: the UML is utilized for requirement description, model specification and design. Then, the UML model is mapped to the PN model for model analysis and simulation, results of which can be fed back to the UML model design. Such process can be executed in an iterative way to achieve continued model improving. Lastly, model implementation is carried out according to the improved UML model.

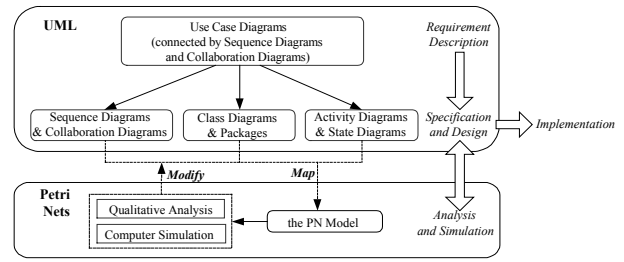


Fig.1: The UML-PNs integrated modeling method

Requirement Description. Firstly, the UML use case diagram - an effective tool for requirement capture and description is used to describe functionality objective of each business activity. Business activities here usually have high abstraction so as not to be trapped into specifics too early. We can connect use cases by sequence diagrams or collaboration diagrams to describe interrelation among functionality objectives. Through use case diagrams users, domain experts, system planners and software designers can be communicated graphically and effectively.

Model Specification and Design. Then, we use multiple UML diagrams to specify and design the static model (by UML class diagrams and packages) and the dynamical model (by UML sequence diagrams, collaboration diagrams, activity diagrams and statechart diagrams) of the business process in the VE.

UML class diagrams are used to describe business objects (including properties and operations) and their associations. In Fig.2, the top-level class architecture of our demonstrative VE is described by the UML class diagram and operations of classes are labeled as P01~P23. As required, this architecture can be extended further level by level with stepwise reification.

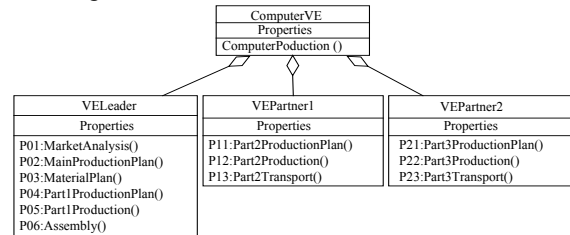


Fig.2: The UML class diagram for business objects

UML packages are used to describe organization patterns of the VE. We can either organize all business objects of a VE member that participate business process execution of the VE into a package or organize business objects belonging to certain domain (e.g., product design, product manufacturing) in the VE into a package.

UML Sequence diagrams and collaboration diagrams are used to describe interactions among business objects. Based on message mechanism, state transitions of business objects are triggered by message transfers, which can be viewed as events (called as *message events* later). Business objects also send messages to themselves, therefore changing states of their own. In Fig.3, message transfers inside/among business objects are described by the UML sequence diagram and message events are labeled as t1~t11.

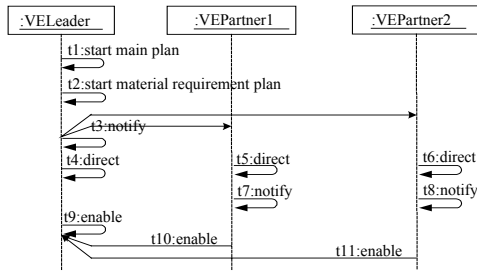


Fig.3: The UML sequence diagram for message transfers

UML activity diagrams are used to describe business activities and their transitions from global view of the whole business process. As business activities correspond to operations in class diagrams, such decomposition actually has been completed after the establishment of class hierarchy. UML activity diagram here concentrates on dynamical relations among business activities. In Fig.4, relations among business activities are described by the UML activity diagram.

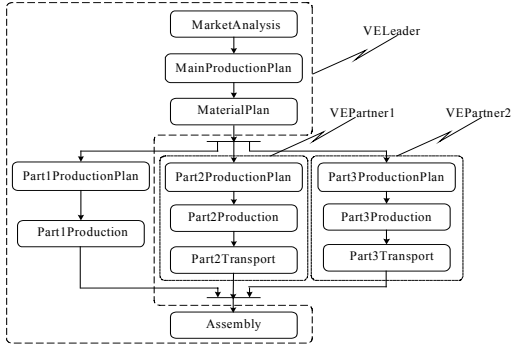


Fig.4: The UML activity diagram for the business process

UML statechart diagrams are used to describe states and transitions inside a business object. States here also correspond to operations in class diagrams. All the states in statechart diagrams of all the objects constitute the set of business activities. Therefore, statechart diagrams describe interrelations of business activities from the object view.

Model Analysis and Simulation. As effective approaches for analyzing the UML model are absent presently, here it is mapped to the PN model for analysis and simulation. The mapping is executed level by level according to the class hierarchy in the UML model. Mapping to sub-PN model at single level follows steps in Tab.1.

Tab.1: The UML-PNs mapping

	The UML	Petri nets
Step 1	Operations in class diagrams (states/ activities in state/ activity diagrams)	Places(P)
Step 2	Message events in sequence diagrams and collaboration diagrams	Transitions(t)
Step 3	Activities/States transitions in Activity diagrams and statechart diagrams	Connections between places and transitions

Fig.5 shows the top-level sub-PN model of our demonstrative VE mapped from UML diagrams in Fig.2~4. Places P01~P23 and transitions t1~t11 correspond to operations P01~P23 in Fig.2 and message events t1~t11 in Fig.3, respectively.

With mapping completion at various abstract levels, a hierarchical, object-oriented PN model with multi-level architecture and stepwise concretion is established. The PN model can be divided into several parts corresponding to business objects, as shown in Fig.5.

